



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/903,726	07/12/2001	John Mark Hartel	AUS920010186US1	9323
7590	07/16/2004			
Duke W. Yee Carstens, Yee & Cahoon, LLP P.O. Box 802334 Dallas, TX 75380			EXAMINER VU, TUAN A	
			ART UNIT 2124	PAPER NUMBER

DATE MAILED: 07/16/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

## Office Action Summary

Application No.

09/903,726

Applicant(s)

HARTEL ET AL.

Examiner

Tuan A Vu

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 12 July 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-27 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-27 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 12 July 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date 2001/09/17.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date: \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_

Art Unit: 2124

### DETAILED ACTION

1. This action is responsive to the application filed July 12, 2001.

Claims 1-27 have been submitted for examination.

#### *Claims objections*

2. Claims 10 and 23 are objected to because of the following informalities: there appears to be a missing ‘,’ between the elements ‘properties’ and ‘comprising’ as recited in line 2. The addition of the ‘,’ in between those elements would help the sentence to be more semantically correct. Appropriate correction is required.

#### *Claim Rejections - 35 USC § 101*

3. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claim 23-26 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a “useful, concrete, and tangible result” be accomplished. An “abstract idea” when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the “useful arts” when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a “useful, concrete and tangible result”.

**As per claim 23**, recited is an interface for defining metadata about a set of properties, such interface comprising a ‘first object’ and a ‘second object’ each describing some metadata. The claim only provides description of some structural elements and does not provide a form of function using or associating those descriptive elements in order to accomplish a result to fulfill

Art Unit: 2124

the purpose of defining metadata using an interface as claimed on the outset. In other words, nowhere is described a set of actions that make use of, or inter-relate the structures described as 'first object' and 'second object' in a context of an interface being applied in some useful arts. Hence, absent any action leading to a concrete, tangible and useful result, the claim fails the requirements of the practical application test. It therefore merely amounts to an abstract idea; thus is rejected for leading to a non-statutory subject matter.

The dependent claims 24-26 do not recite any structure or functions that cure the deficiency of claim 23, and therefore are also rejected for reciting non-statutory subject matter.

***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1-22, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sun Microsystems, "Version 1.1.8 of Java Platform API Specification", 1995-1999 ( hereinafter Jdk118), in view of Heirstermann et al., USPN: 6,477,701 (hereinafter Heirstermann).

**As per claim 1**, Jdk118 discloses a method for providing metadata, comprising:

creating an object descriptor class for an object (e.g. java.beans.BeanDescriptor, pg. 1 of

2 – Note: object created as from a constructor for 'BeanDescriptor ( class, class)' is object instantiating said Descriptor class being construed for instantiation);

providing an object descriptor interface using the object descriptor class (Note: constructor for creating an instance of Descriptor Api object inherently teaches object descriptor interface class using its instantiated object, the object descriptor class)

defining methods defined by the object descriptor interface to identify a property of the object (e.g. java.beans.BeanDescriptor : *Methods* - pg. 3 of 3; java.beans.BeanDescriptor pg. 2 of 2; java.beans.FeatureDescriptor, pg. 2,3,4 of 4 – Note: the BeanDescriptor and FeatureDescriptor interface being implemented by an object, e.g. in BeanDescriptor ( class, class) is equivalent to being created and associated for identification of a property ,or attributes, name, value related to the object class);

creating a property descriptor class for the property of the object (e.g. java.beans.PropertyDescriptor: *Constructor Index* - Note: object created as from constructor 'PropertyDescriptor ( string, class)' is equivalent to existence of the class PropertyDescriptor being constructed to yield the object instance to obtain property information of the object);

providing a property descriptor interface using the property descriptor class (Note: constructor for creating an instance of propertyDescriptor Api inherently teaches interface class for using propertyDescriptor object being instantiated ); and

defining methods defined by the property descriptor interface associated with the property (e.g. java.beans.PropertyDescriptor: *Method Index*, pg. 3,4 of 4).

But Jdk118 does not expressly teach using methods that have been defined by the object descriptor interface to identify a property, or using the methods defined by the property descriptor interface to obtain metadata associated with the property. Jdk118 teaches remote procedure calls and stream serializing in conjunction with descriptors (e.g. Class Descriptors,

Art Unit: 2124

*objectStreamClass descriptor*- pg. 1 of 2; *Server Interfaces, FileDescriptor* - pg. 4 of 9).

Creating class or interface methods in an object-oriented programming language for an application purpose was a known concept and using application interface (e.g. jdk1.1.8 api) to retrieve metadata or property information on class-derived objects using descriptor class API was also a known concept at the time the invention was made. Heistermann, in a method using API similar to Jdk118's interface associating a descriptor class or a property descriptor class to interface methods, discloses the use of such API methods to identify a property related to a object derived from a descriptor class or obtain metadata associated with such property ( e.g. col. 7, line 9 to col. 8, line 9; Fig. 9, 11). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide the methods as created by Jdk118 so that the methods can be utilized according to the suggested application by Heistermann, to identify a property and to obtain metadata thereof. One skill in the art would be motivated to provide such a use as taught by Heistermann to the API/ interface methods for object descriptor and for the property descriptor so that property and additional information on those properties can be gathered in order to achieve identification and verification of objects as they are arrived from serialized transmission format, e.g. as suggested in Jdk118 teachings, such that compatibility (e.g. version) of those objects as they are gathered for application execution would be secured according to one aspect of usage as mentioned by Heistermann ( e.g. col. 6, lines 1-17).

**As per claim 2**, Jdk118 discloses object descriptor interface extending a property set descriptor interface (e.g. java.beans.BeanDescriptor: *extends FeatureDescriptor* - pg. 1 of 2 – Note: FeatureDescriptor being a base interface for other descriptor interface children is

Art Unit: 2124

equivalent to property set descriptor interface being extended by other object descriptor or property descriptor interfaces)

**As per claim 3**, Jdk118 discloses the property set containing the property descriptor interface (java.beans.FeatureDescriptor, ...*common base class for PropertyDescriptor* - pg. 1 of 4) Note: FeatureDescriptor is a base interface class and contains a descriptor interface which extends such base interface, e.g. PropertyDescriptor api )

**As per claim 4**, Jdk118 discloses single property (e.g. java.beans.PropertyDescriptor: Methods, isBound() – pg. 3 of 4).

**As per claim 5**, Jdk118 discloses collection of property descriptions into a single object description (e.g. java.beans.BeanDescriptor : *Methods* - pg. 3 of 3; java.beans.BeanDescriptor pg. 2 of 2; java.beans.FeatureDescriptor: Method Index – Note: the combination of methods (inherited) from parent interface class, FeatureDescriptor, and child interface class, BeanDescriptor, is equivalent to collection of properties into a single child object property description)

**As per claim 6**, Jdk118 does not explicitly disclose that the object can be a database object. However, Jdk118 teaches RMI calls and descriptor used in stream serializing ( re claim 1). Further, the use of beans to remote communicating with a COM or CORBA interface in order to effect remote calls to dababase was a known concept in the use of server beans at the time the invention was made. Official notice is taken that the use of beans ( e.g. enterprise beans) for effecting remote objects retrieval from databases via a server operable with a COM or ORB interface was a known concept at the time the invention was made. Hence, based on the suggestion by Jdk118 and teachings by Heistermann to create descriptor for identifying streamed

Art Unit: 2124

data, in view of known practice to query database via beans and stream communication via remote calls as suggested above, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide a database object ( such as suggested by common beans/ORB services) as target of property gathering via a Descriptor interface such as suggested by Heistermann or Jdk118. This database object for which a Descriptor object and PropertyDescriptor interface are being implemented would have been obvious in light of the above common teachings combined with Jdk118 and Heistermann because data retrieved from remote storage can undergo compatibilities or time frame problems as mentioned by Heistermann; and creating APIs for investigating the properties of objects being retrieved from database would enhance fault-free data utilization and also alleviate incompatibilities issues coming from stream of data for the reasons set forth in claim 1.

**As per claim 7**, the limitation as to use methods in Descriptor interface to identify a property of object using API calls to the database fall under the ambit of the limitation of claim 6 and would have been also obvious for the same reasons as set forth therein.

**As per claim 8**, Jdk118 discloses java Beans (e.g. java.beans.BeanDescriptor; java.beans.PropertyDescriptor)

**As per claim 9**, even though Jdk118 discloses mechanism for some Descriptor class (e.g. java.beans.PropertyDescriptor, Constructors: ...IntrospectionException – pg. 2 of 4; java.beans.IndexedPropertyDescriptor, Constructors: ...IntrospectionException – pg. 2 of 3), Jdk118 implicitly teaches of such mechanism to all object Descriptor interfaces from a same parent -- Note: the methods within an Descriptor class descendant of a base FeatureDescriptor class inherit the same exception methods from the base class).



**As per claim 10**, Jdk118 disclose a method for providing metadata about a set of properties, comprising:

creating a first object describing metadata for a plurality of data sources, wherein the first object stores a related set of properties (e.g. java.beans.BeanDescriptor, pg. 1 of 2 - Note: object created as from a constructor for BeanDescriptor ( class, class) encompasses metadata or data source - each property being defined for the object being a source -- concerning the object being described via the Descriptor object instantiation); and

creating a set of second objects describing metadata for respective properties of the data sources (e.g. java.beans.PropertyDescriptor – Note: interfaces coming from a base interface, e.g. java.beans.FeatureDescriptor, are equivalent to a set of second objects); wherein metadata concerning a respective property is obtained from method in a respective one of the second objects (e.g. java.beans.BeanDescriptor : *Method Index*, pg. 3,4 of 4).

But Jdk118 does not expressly teach that dynamically defined metadata is obtained by invoking a method in one respective second objects. Using of beans and related APIs as defined by Jdk118 to effect properties identification and retrieval was a known concept in server application and remote procedure calls addressing data stored in database or via interface with COM or ORB. And dynamic or asynchronous data stream handling has been suggested by either Jdk118 (Jdk118 teaches remote procedure calls and stream serializing in conjunction with descriptors (e.g. Class Descriptors, *objectStreamClass descriptor*- pg. 1 of 2; Server Interfaces, *FileDescriptor* - pg. 4 of 9) or Heistermann ( re claim 1); hence the concept of dynamic defining of metadata being invoked by beans APIs is implied. Besides, creating methods in an object-oriented class or interface for a purpose or application use was a known concept in programming

Art Unit: 2124

language and using application interface (e.g. jdk1.1.8 api) to retrieve metadata or property information on derived objects using descriptor class API was also a known concept at the time the invention was made. Thus, it would have been obvious for one of ordinary skill in the art at the time the invention was made to use the methods as defined in the Descriptor object( First object) or PropertyDescriptor object ( second object) as taught by Jdk118 so that dynamic metadata being defined via invoking those methods can be effected just as suggested by known concepts (bean API and remote calls to database/Com) or Heistermann's teachings ( re claim 1). The reasons for this would be the same as set forth in claim 1 above using Heistermann's approach to alleviate fault in handling serialized data stream and the benefits from common practices as mentioned above using beans (e.g. Jdk118 serialized and RMI teachings) service to dynamically create metadata information on retrieved data via remote calls.

**As per claim 11**, Jdk118 teaches that the first object and set of second objects comprise interface objects (e.g. java.beans.BeanDescriptor, pg. 1 of 2; java.beans.PropertyDescriptor – Note: object instantiated from an API like BeanDescriptor or PropertyDescriptor are equivalent to interface objects of 1<sup>st</sup> and 2<sup>nd</sup> object respectively)

**As per claim 12**, this claim amounts to the limitation of claim 4 above, i.e. the second object being defined with methods to describe a single property (e.g. refer to java.beans chapter on PropertyDescriptor: Methods, isBound() – pg. 3 of 4)

**As per claim 13**, this claim corresponds to claim 5 and is rejected using the corresponding rationale accordingly.

**As per claim 14**, this claim is the apparatus version claim of claim 1 above, and includes implementation means for steps of

Art Unit: 2124

creating (object descriptor);

providing (object descriptor interface);

using (methods ... object descriptor interface);

creating (property descriptor);

providing (property descriptor interface);

using (methods ... property descriptor interface) just as recited in claim 1. Hence these step limitations are rejected with the corresponding rejection as set forth in claim 1, respectively.

**As per claims 15-22**, these claims correspond to claims 2-9, respectively; and are rejected using the corresponding rejection as set forth therein, respectively.

**As per claim 27**, this is a computer-readable medium version of claim 1, and includes instructions for performing the same steps limitations as recited therein; hence is rejected with the corresponding rejections addressing those limitations therein as have been set forth above.

***Claim Rejections - 35 USC § 102***

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 23-26 are rejected under 35 U.S.C. 102(b) as being anticipated by Sun Microsystems, “Version 1.1.8 of Java Platform API Specification”, 1995-1999 ( i.e. Jdk118).

**As per claim 23**, Jdk118 discloses a method for providing metadata about a set of properties, comprising:

Art Unit: 2124

creating a first object describing metadata for a plurality of data sources, wherein the first object stores a related set of properties (e.g. java.beans.BeanDescriptor, pg. 1 of 2 - Note: object created as from a constructor for BeanDescriptor ( class, class) encompasses metadata or data source - each property being defined for the object being a source -- concerning the object being described via the Descriptor object instantiation); and

creating a set of second objects describing metadata for respective properties of the data sources (e.g. java.beans.PropertyDescriptor - Note: interfaces coming from a base interface, e.g. java.beans.FeatureDescriptor, are equivalent to a set of second objects); wherein metadata concerning a respective property is obtained from method in a respective one of the second objects (e.g. java.beans.BeanDescriptor : *Method Index*, pg. 3,4 of 4).

**As per claim 24**, Jdk118 teaches that the first object and set of second objects comprise interface objects (e.g. java.beans.BeanDescriptor, pg. 1 of 2; java.beans.PropertyDescriptor - Note: object instantiated from an API like BeanDescriptor or PropertyDescriptor are equivalent to interface objects of 1<sup>st</sup> and 2<sup>nd</sup> object respectively)

**As per claim 25**, this claim amounts to the limitation of claim 4 above, i.e. the second object being defined with methods to describe a single property of the data source(e.g. refer to java.beans chapter on PropertyDescriptor: Methods, isBound() - pg. 3 of 4)

**As per claim 26**, this claim corresponds to claim 5 and is rejected using the corresponding rationale accordingly.

### ***Conclusion***

Art Unit: 2124

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

**Any response to this action should be mailed to:**

Commissioner of Patents and Trademarks

Washington, D.C. 20231

**or faxed to:**

(703) 872-9306 ( for formal communications intended for entry)

**or:** (703) 746-8734 ( for informal or draft communications, please label

"PROPOSED" or "DRAFT" – please consult Examiner before use)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington, VA. , 22202. 4<sup>th</sup> Floor( Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

VAT  
July 11, 2004

*Kakali Chaki*  
KAKALI CHAKI  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100